

## **REMARKS**

Claims 1-34 are pending in the application. Claims 1-34 stand rejected under 35 U.S.C. § 103(a) over Burns in view of various secondary references. In view of the following remarks, reconsideration and withdrawal of these grounds of rejection is requested.

### **Claim Rejections Under 35 U.S.C. § 103**

Claims 1-4, 6-17, 19 and 21-34 stand rejected under 35 U.S.C. § 103(a) as being obvious over Burns et al. (U.S. Pat. No. 5,931,947) in view of Hsu (U.S. Pat. No. 5,584,023). In view of the following remarks, reconsideration and withdrawal of this ground of rejection is respectfully requested.

The present invention comprises a method for accessing and protecting files in a Windows-based computer operating system. The method checks for “spoofed” files each time a file system request is made. A “spoofed” file is a file which has been secured by placing the file data in a secured location where it is not readily accessible by a user. The user interfaces with the “spoofed” (secured) file through a tag file (preferably of zero (0) bytes) which is accessible to the user, and which appears to the user to be the entire “spoofed” file. In reality the tag file does not contain all the information of the “spoofed” file, but merely identifies the actual (secure) location of the data from the “spoofed” file.

Claim 1 recites:

A method for providing data security in a device driver for accessing data, the method comprising the steps of: providing at least one tag file, the tag file comprising a physical file of zero bytes in length; maintaining the tag file in a part of a file system; and processing file system calls so that the tag file appears as a secured file containing data from the view point of a user, operating system and programs, the method further comprising the steps of: detecting a file system request; completing said file system request; receiving return information from said file system request; determining whether said file system request is for a tag file associated with a secured file; and if so, modifying said return information to reflect a file attribute of the secured file. [emphasis added].

Thus, claim 1 requires a method for providing data security including the steps of “providing at least one tag file...of zero bytes,” “maintaining the tag file,” “processing file system calls so that the tag files appears as a secured file...”, and “determining whether [a] file system request is for a tag file associated with a secured file.” As explained below, neither Burns nor Hsu discloses, teaches or suggests such an invention.

Burns discloses a network storage device 1 coupled to a network 2, and a plurality of network clients (see, col. 5, lines 60-65). By the Examiner’s own admission, Burns fails to disclose, teach or suggest “at least one tag file” or any of the other limitations in claim 1 relating to the “tag file.”

Hsu teaches a computer system 10 which provides encryption of data through a Central Processor Unit (CPU) 12 (See Fig. 1; col. 5, line 443 – col. 6, line 19). The CPU 12 may operate in one of two modes, “user” mode and “kernel” mode (see, col. 5, lines 59-62). The user may access system files through an application program 26 existing in the user mode which makes requests (calls) to a trap handler 32 existing in the kernel mode (See Fig. 2; col. 6, lines 1-19). The possible system requests are represented in a system entry (“sysent”) table 34 existing in the kernel mode and coupled to the trap handler 32. These system request entries include the functions of OPEN, CREATE, READ, WRITE, CHMOD (Change Mode), FORK, STATF (Status), SEEK, EXIT and IOCTL (Input/Output Control) (see, col. 6, lines 33-43).

The OPEN and CREATE procedures creates or accesses “inode” information for the particular file, which comprises information describing the protection mode of the file, owner, user group and size of the file (see, col. 6, lines 44-57). Hsu explains that a user may request a file be opened (through an OPEN request) or created (through a CREATE request), and then immediately truncated to zero file length (see, col. 17, lines 39-56). If such a file (to be opened or created) is encrypted, the system will require authentication of the user before performing such a truncation function (see, col. 17, lines 39-56).

Hsu fails to disclose, teach or suggest a “tag file” of zero bytes which may be associated with a “secured file,” as recited in claim 1. The file which Hsu explains may be “opened and immediately truncated to zero file length” is not a tag file and is not associated with any secured file. The file to which Hsu refers is a system file which a user has selected to ‘open’ or ‘create,’ and truncate to zero file length. Even if, arguendo, the file described by Hsu could be considered

a “tag file” (and it cannot), the file is not associated with a “secured file” as required by claim 1. In sum, the file described by Hsu is not a “tag file”, is not associated with (i.e., specifies the location of) a “secured file,” as claim 1 requires. Accordingly, reconsideration and withdrawal of this ground of rejection with respect to claims 1-4 and 6-14 is respectfully requested.

Independent claims 15, 30 and 32 recite similar limitations to those discussed above with reference to independent claim 1. In particular, all of claims 15, 30 and 32 recite a “tag file” which is associated with a “secured file.” Because, as explained above, neither Burns nor Hsu disclose, teach or suggest a “tag file” associated with a “secured file” reconsideration and withdrawal of this ground of rejection with respect to claims 15-17, 19 and 21-34 is respectfully requested.

Claims 5 and 20 stand rejected under 35 U.S.C. § 103(a) as being obvious over Burns et al. in view of Hsu, and further in view of Johnson et al. (U.S. Pat. No. 4,887,204). In view of the following remarks, reconsideration and withdrawal of this ground of rejection is respectfully requested.

As discussed above, neither Burns nor Hsu disclose, teach or suggest a “tag file” associated with a “secured file.” Johnson also fails to disclose, teach or suggest such an invention.

Johnson teaches a system and method for accessing remote files in a network. Johnson also teaches a virtual file system (See Fig. 10). Johnson fails to disclose, teach or suggest a “tag file” associated with a “secured file,” as required by independent claims 1 and 15, from which claims 5 and 20 depend. Therefore, reconsideration and withdrawal of this ground of rejection is respectfully requested.

Claim 18 stands rejected under 35 U.S.C. § 103(a) as being obvious over Burns et al. in view of Hsu, and further in view of Roberts (U.S. Pat. No. 5,287,453). In view of the following remarks, reconsideration and withdrawal of this ground of rejection is respectfully requested.

As discussed above, neither Burns nor Hsu disclose, teach or suggest a “tag file” associated with a “secured file.” Roberts also fails to disclose, teach or suggest such an invention.

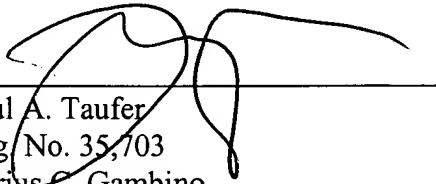
Roberts teaches a Fast Remote File Access (FRFA) system which includes a file manager module 12-240 which translates requests from call handler 12-140 into requests or messages

(See Fig. 2). Roberts fails to disclose, teach or suggest a "tag file" associated with a "secured file," as required by independent claim 15, from which claim 18 depends. Accordingly, reconsideration and withdrawal of this ground of rejection is respectfully requested.

### **Conclusion**

In view of the foregoing remarks, Applicants submit that this application is in condition for allowance at an early date, which action is earnestly solicited.

Respectfully submitted,



---

Paul A. Taufer  
Reg. No. 35,703  
Darius C. Gambino  
Reg. No. 41,472

Piper Rudnick LLP  
One Liberty Place  
1650 Market Street, Suite 4900  
Philadelphia, PA. 19103  
Phone: 215.656.3300